

# Analizzatore di pacchetti per test di regressione su protocollo eBUS tramite integrazione del linguaggio di scripting LUA realizzato in NI LabVIEW

Davide Faccio - IPSES

## LA SFIDA

Realizzare un tool in grado di identificare le violazioni delle regole previste dal protocollo di comunicazione eBUS analizzando i file di log di una tipica sessione di funzionamento di una rete di installazione di N device intelligenti. Come requisito fondamentale del sistema è richiesto un metodo flessibile per la stesura delle regole del protocollo, acquisti in tempo reale per determinare la conformità del prodotto alle specifiche richieste.

Tra gli apparati per il comfort e il riscaldamento in ambito domestico, da alcuni anni, è presente un nuovo protocollo di comunicazione chiamato eBUS tramite il quale i device presenti all'interno di una stessa rete sono in grado di interagire tra loro con un discreto grado di intelligenza in modo da fornire o richiedere servizi in un contesto di stabilità e corretto funzionamento dell'impianto.

Ovviamente, un requisito fondamentale è che tali device comunichino nel rispetto delle regole previste dal protocollo, inoltre, a causa dell'elevato grado di flessibilità ed espandibilità che può avere l'installazione (N device, M zone di interesse etc.), il BUS di comunicazione può diventare molto affollato e può quindi essere molto complicato verificare il comportamento globale delle procedure di comunicazione all'interno della rete. Per queste ragioni, in fase di rilascio di nuovi prodotti, è molto importante eseguire i test necessari a verificare il rispetto delle regole dell'eBUS quando i device sono immersi in una rete reale creata dall'impianto.

La sfida è stata quella di realizzare un tool che fosse in grado di analizzare alcuni file di log contenenti lo storico delle comunicazioni avvenute su BUS durante tipiche sessioni di funzionamento dell'impianto. Tale strumento deve essere in grado di identificare le violazioni delle regole previste dal protocollo eBUS e dedurre così eventuali avarie o malfunzionamenti dei device coinvolti. Le regole previste dal protocollo eBUS possono essere semplici o articolate a seconda del campo di applicazione. Alcune regole riguardano la

## LA SOLUZIONE

È stato realizzato un tool tramite NI LabVIEW in grado di analizzare i dati dei file di log e identificare le violazioni delle regole previste dall'eBUS sfruttando un set di script per linguaggio LUA. Tale tecnologia di scripting è stata integrata nel tool per rendere il set di regole customizzabile da parte dell'utente finale senza ricompilazione del tool di analisi.

natura del pacchetto trasmesso o ricevuto dai device, altre sono relative a procedure stratificate di fasi temporali attraverso cui l'impianto deve passare per arrivare alla stabilità (ad esempio la fase di start up o la contesa dei servizi). In quest'ultimo caso le regole possono essere notevolmente complicate e strutturate in sotto regole o requisiti interconnessi tra loro. Proprio a causa di questa complessità e al fatto che le analisi richieste sono soggette ad aggiornamenti e a continue aggiunte, è stato necessario proporre una tecnologia che risultasse flessibile e customizzabile dal cliente stesso. Per tale motivo si è deciso di utilizzare un linguaggio di scripting per la stesura delle regole.

Da ormai molti anni nel panorama informatico i linguaggi di scripting ad alta astrazione e interpretati affiancano e coadiuvano i linguaggi compilati o precompilati in modo da rendere più flessibili i prodotti software finali in termini di customizzazione post compilazione. Per la nostra applicazione l'attenzione si è rivolta verso un veloce e potente linguaggio di scripting chiamato LUA (Open

*“È stato realizzato un tool tramite NI LabVIEW in grado di analizzare i dati dei file di log e identificare le violazioni delle regole previste dall'eBUS.”*

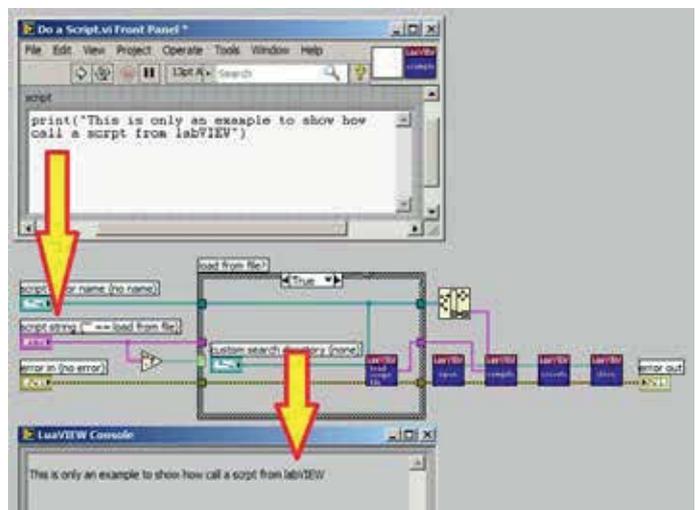


Figura 1: Esempio di chiamata di un LUA script. La stringa o il file di testo vengono compilati ed eseguiti tramite VI del pacchetto LuaVIEW. Il risultato viene stampato sulla console

Source con licenza MIT) in grado di interfacciarsi a LabVIEW grazie al pacchetto LuaVIEW per il quale è richiesta la LabVIEW run-time license.

L'architettura pensata per il nostro tool prevede l'utilizzo di LabVIEW per la costruzione dell'apparato software che gestisce lo scripting engine, il caricamento dati, la creazione delle strutture usate dagli script, la user Interface e parte degli algoritmi previsti da alcune funzioni usate negli script.

In merito al pacchetto LuaVIEW è di fondamentale importanza per il fatto che esso può effettuare chiamate da LabVIEW verso LUA e viceversa: infatti, nel primo caso, è stato realizzato lo scripting engine in grado di processare in parallelo un numero variabile di task corrispondenti alle diverse regole del protocollo (vedi esempio figura 1), nel secondo caso è stato possibile creare funzioni di LabVIEW usate all'interno degli script per ottenere e manipolare i dati precedentemente caricati in memoria (vedi figura 2).

L'interazione con LabVIEW avviene attraverso alcune API fornite dal pacchetto LuaVIEW per inserire e eliminare (pull and push) dati nello stack di LUA. Tali API consentono il passaggio dei parametri con LabVIEW anche nel caso di tipi di dati non semplici, come cluster o array di cluster. Inoltre, permettono di creare funzioni parametrizzate sotto forma di VI richiamabili da LUA. In questo modo si riesce a bilanciare la creazione degli algoritmi che descrivono le regole del BUS tra il mondo LUA e il mondo LabVIEW.

Sebbene LUA non sia propriamente un linguaggio Object Oriented, di fatto permette di costruire strutture e metodi simulando il comportamento tipico della programmazione con le classi e, in questo senso, rende la stesura degli script molto potente ed efficace.

Il caricamento dei file di log popola una struttura dati che verrà costantemente utilizzata negli script per verificare le violazioni delle regole. Ogni regola è creata in modo da fornire una result table con cui è possibile identificare i pacchetti interessati dalle violazioni e, tramite l'interfaccia grafica, visualizzare i frame o i blocchi di frame coinvolti in tali violazioni.

Le regole, implementate sotto forma di script, sono di fatto file di testo e, in quanto tali, possono essere create, modificate e cancellate dall'utente finale in modo da plasmare l'analisi secondo esigenze puntuali, pur rimanendo svincolati dalla necessità di una compilazione del tool a fronte di modifiche introdotte nelle regole.

## Risultati

Grazie all'aggiunta di una tecnologia basata su scripting language si sono estese enormemente le potenzialità già offerte da LabVIEW, riuscendo a ottenere un tool estremamente flessibile e configurabile, con notevoli vantaggi in termini di riduzione dei costi e tempistiche di sviluppo.

## Prodotti utilizzati:

LabVIEW

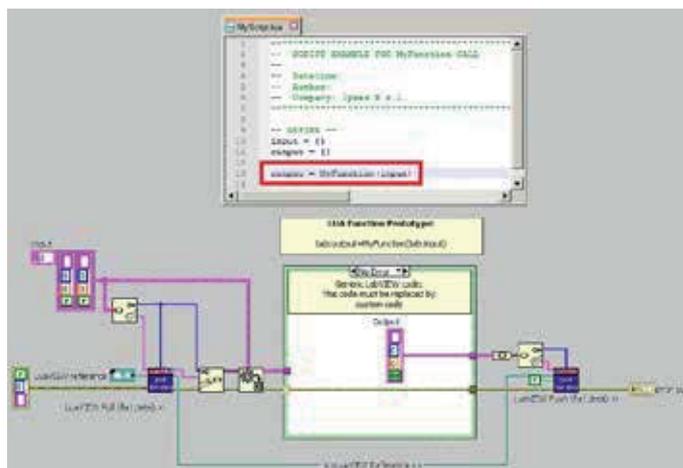


Figura 2: Lo script LUA contiene la chiamata alla funzione MyFunction. Input e output sono dei tipi di dato generico scambiati tra la libreria LUA e LabVIEW tramite le API. Il codice LabVIEW implementa il core della funzione